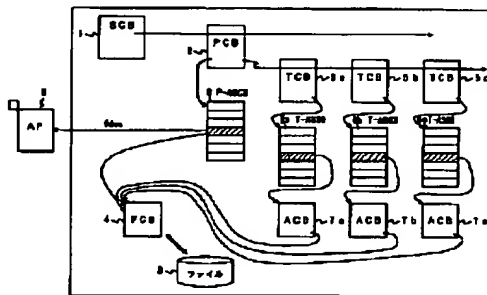


(43) Date of publication of application: 11 . 03 . 97

(72) Inventor: IDE SHUNICHI

**(57) Abstract:**

**SOLUTION:** When a process 9 makes a request to open a file 8, a file descriptor is assigned to the file 8 and sent back to the process 9 at the request source; when some thread of the process 9 makes a request to access the file 8 for the 1st time by using the file descriptor, access control information regarding the file 8 including a pointer is held by access control blocks 7a-7c. The access control blocks are secured on a memory while made to correspond to the thread, and then the file descriptor is handled as a resource in process units; and threads have the access control blocks 7a-7d individually and then threads in the same process use the file descriptor not exclusively.



COPYRIGHT: (C)1997,JPO

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平9-69060

(43) 公開日 平成9年(1997) 3月11日

(51) Int.Cl.<sup>6</sup>  
G 0 6 F 12/00

識別記号  
5 3 5

庁内整理番号

F I  
G 0 6 F 12/00

技術表示箇所

5 3 5 Z

審査請求 未請求 請求項の数2 O L (全 7 頁)

(21) 出願番号 特願平7-223824

(22) 出願日 平成7年(1995) 8月31日

(71) 出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72) 発明者 井手 俊一

東京都青梅市末広町2丁目9番地 株式会  
社東芝青梅工場内

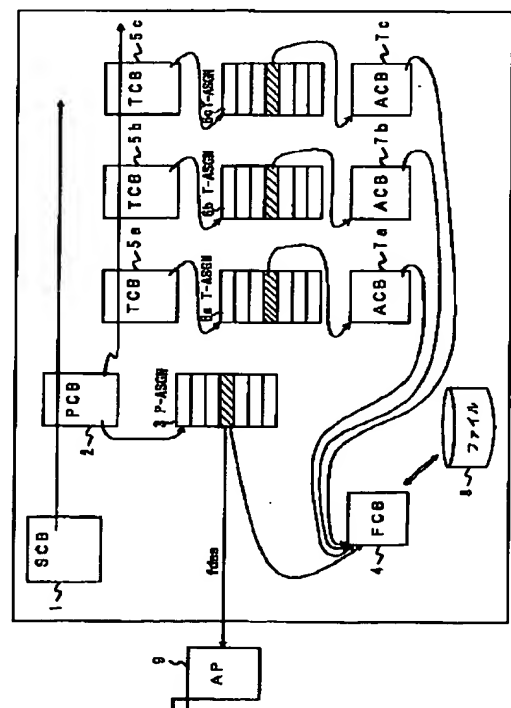
(74) 代理人 弁理士 鈴江 武彦

(54) 【発明の名称】 計算機システムおよびファイルアクセス制御方法

(57) 【要約】

【課題】 ファイル記述子をプロセス単位の資源として扱うとともに、このファイル記述子を排他せずに使用することのできる計算機システムを提供する。

【解決手段】 プロセス9がファイル8のオープンを要求したときに、ファイル9にファイル記述子を割り当て、そのファイル記述子を要求元のプロセス9に返送し、プロセス9のいずれかのスレッドがそのファイル記述子を用いてファイル8へのアクセスを初めて要求したときに、ポインタを含むファイル8に関するアクセス制御情報を保持するアクセス制御ブロック7a~7cをそのスレッドに対応させてメモリに確保することにより、ファイル記述子をプロセス単位の資源として扱うとともに、スレッドそれぞれが個別にアクセス制御ブロック7a~7cをもつことによって、同一プロセス内のスレッドすべてがファイル記述子を排他することなく使用することを特徴とする。



## 【特許請求の範囲】

【請求項 1】 一つのプロセスが CPU への割り付け単位となるスレッドを複数もつことのできるマルチスレッド環境を提供する計算機システムであって、前記スレッドそれぞれが、ファイルそれぞれに固有に割り当てられたファイル記述子を用いてファイルへのアクセスを行なう計算機システムにおいて、

前記プロセスがファイルのオープンを要求したときに、そのオープンの要求されたファイルにファイル記述子を割り当てて、そのファイル記述子を前記要求元のプロセスに返送する手段と、

前記プロセスのいずれかのスレッドが前記割り当てられたファイル記述子を用いてそのファイルへのアクセスを初めて要求したときに、アクセスポインタを含むそのファイルに関するアクセス制御情報を保持するアクセス制御ブロックをそのスレッドに対応させて前記メモリに確保する手段とを具備し、

前記ファイル記述子をプロセス単位の資源として扱うとともに、前記スレッドそれぞれが個別にアクセス制御ブロックをもつことによって、同一プロセス内のスレッドすべてが前記ファイル記述子を排他することなく使用することを特徴とする計算機システム。

【請求項 2】 一つのプロセスが CPU への割り付け単位となるスレッドを複数もつことのできるマルチスレッド環境を提供する計算機システムであって、これらのスレッドそれぞれがファイルそれぞれに固有に割り当てられたファイル記述子を用いてファイルへのアクセスを行なう計算機システムのファイルアクセス制御方法において、

前記プロセスがファイルのオープンを要求したときに、そのオープンの要求されたファイルにファイル記述子を割り当てて、そのファイル記述子を前記要求元のプロセスに返送するステップと、

前記プロセスのいずれかのスレッドが前記割り当てられたファイル記述子を用いてそのファイルへのアクセスを初めて要求したときに、アクセスポインタを含むそのファイルに関するアクセス制御情報を保持するアクセス制御ブロックをそのスレッドに対応させて前記メモリに確保するステップとを具備し、

前記ファイル記述子をプロセス単位の資源として扱うとともに、前記スレッドそれぞれが個別にアクセス制御ブロックをもつことによって、同一プロセス内のスレッドすべてが前記ファイル記述子を排他することなく使用することを特徴とするファイルアクセス制御方法。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】 本発明は、マルチスレッド環境を提供する計算機システムおよび同計算機システムに適用して好適なファイルアクセス制御方法に係り、特にファイル記述子をプロセス単位の資源として扱うととも

に、同一プロセス内のスレッドすべてがファイル記述子を排他することなく使用することを可能とする計算機システムおよびファイルアクセス制御方法に関する。

## 【0002】

【従来の技術】 近年の計算機システムの普及は目覚ましいものがあり、また、この普及に伴って処理の高速化がますます要求されてきている。そして、この高速化を実現するものの一つとして、複数のプロセッサを有する並列計算機システムが存在し、この並列計算機システム上で主に提供される環境としてマルチスレッド環境が存在する。

【0003】 このマルチスレッド環境で実行されるプロセス、すなわちアプリケーションは、プロセッサへの割り付け単位となるスレッドを複数もつことが可能であり、したがって、同一プロセス内のスレッドが別々のプロセッサで並列に実行することができるため、処理の高速化が図られることになる。

【0004】 ここで、図 6 を参照して従来のマルチスレッド環境下でのファイルアクセスの制御原理について説明する。プロセス (AP) 17 が生成されたとき、システムは、プロセス制御ブロック (PCB) およびアサインテーブル (ASGN) をメモリに確保して、このプロセス制御ブロックをシステム制御ブロック (SCB) 11 に登録する。ここでは、プロセス制御ブロック 12a およびアサインテーブル 13 がプロセス 17 に対応して確保されたものとする。

【0005】 このシステム制御ブロック 11 は、システム上で稼働するプロセス全体を制御するために用いられるものである。また、プロセス制御ブロック 12a ~ 12c およびアサインテーブル 13 は、生成されたプロセスそれぞれに確保されるものである。そして、プロセス制御ブロックは、そのプロセスの状態管理などに用いられるものであり、アサインテーブルは、後述するアクセス制御ブロック (ACB) を管理するために用いられるものである。

【0006】 ここで、このプロセス 17 がファイル 16 のオープンを要求すると、システムは、そのファイル 16 に対応させてアクセス制御ブロック 14 をメモリに確保して、このアクセス制御ブロック 14 をアサインテーブル 13 のいずれかのエンTRIES に設定する。そして、この設定したエンTRIES をファイル記述子 (fdes) としてプロセス 17 に返送する。

【0007】 このアクセス制御ブロック 14 には、有編成ファイルにおけるアクセスポインタやレコードロック情報、無編成ファイルにおけるカレントアクセス位置などといったアクセス制御情報が保持される。なお、このようなシステムでは、ファイルそれぞれにファイル制御ブロック (FCB) を必要に応じてメモリに確保し、このファイル制御ブロックにファイルの制御情報を保持している。そして、アクセス制御ブロックは、このファイ

ル制御ブロックを参照するためのアドレスなどを保持している。

【0008】そして、プロセス17のスレッドそれぞれは、前述のオープン時にシステムから受けとったファイル記述子を用いてファイル16へのアクセス要求を発行する。

【0009】これによりシステムは、アクセス制御ブロック14に基づいたファイルアクセスを実施することができる。このように、マルチスレッド環境で実行されるプロセスでは、一つのファイルに対し、各スレッドが同じファイル記述子を用いてアクセスを行なうため、ファイル記述子をプロセス単位の資源（プロセスコンテキスト）として扱うことができ、ファイルのオープンおよびクローズを、プロセス単位の処理としてそれぞれ初期処理および終了処理で一括して行なうなどの構造化プログラミングを可能としていた。

【0010】しかしながら、各プロセスが、アクセス制御ブロックを一つのファイルに対して一つしか確保していないために、スレッド相互間でファイル記述子を排他して用いなければならず、したがって、ファイル記述子の排他を考慮したプログラミングを行なわなければならなかった。

【0011】一方、ファイルのオープンおよびクローズを、プロセス単位の処理として行なうのではなく、各スレッドが、アクセスするファイルについて独自にオープンおよびクローズを行ない、アクセス制御ブロックをスレッドそれぞれに確保するといった方法も存在する。

【0012】しかしながら、ファイルのオープンおよびクローズを、スレッド単位の処理として行なうと、同一ファイルについてファイル記述子が複数設定されてしまうこととなり、ファイル記述子をプロセス単位の資源として扱うことができなくなるために、プログラム構造をいたずらに複雑にしてしまうとともに、その生産性を著しく低下させてしまう。

【0013】

【発明が解決しようとする課題】 前述したように、従来のマルチスレッド環境下におけるファイルアクセス制御では、複数のスレッドが単一のアクセス制御ブロックを使用するために、プロセス内のスレッドそれぞれが、ファイル記述子を排他的に使用しなければならなかった。

【0014】また、複数のアクセス制御ブロックを確保する場合には、スレッドごとに同一ファイルをオープンし、スレッドごとに異なるファイル記述子を使用しなければならないため、ファイル記述子がプロセスコンテキスト、すなわちプロセス単位の資源として扱うことはできなかった。

【0015】本発明は、このような実情に鑑みてなされたものであり、ファイル記述子をプロセス単位の資源として扱うとともに、同一プロセス内のスレッドすべてがファイル記述子を排他することなく使用することを可能

とする計算機システムおよびファイルアクセス制御方法を提供することを目的とする。

【0016】

【課題を解決するための手段】 本発明の計算機システムは、一つのプロセスがCPUへの割り付け単位となるスレッドを複数もつことのできるマルチスレッド環境を提供する計算機システムであって、前記スレッドそれぞれが、ファイルそれぞれに固有に割り当てられたファイル記述子を用いてファイルへのアクセスを行なう計算機システムにおいて、前記プロセスがファイルのオープンを要求したときに、そのオープンの要求されたファイルにファイル記述子を割り当てて、そのファイル記述子を前記要求元のプロセスに返送する手段と、前記プロセスのいずれかのスレッドが前記割り当てられたファイル記述子を用いてそのファイルへのアクセスを初めて要求したときに、アクセスポイントを含むそのファイルに関するアクセス制御情報を保持するアクセス制御ブロックをそのスレッドに対応させて前記メモリに確保する手段とを具備し、前記ファイル記述子をプロセス単位の資源として扱うとともに、前記スレッドそれぞれが個別にアクセス制御ブロックをもつことによって、同一プロセス内のスレッドすべてが前記ファイル記述子を排他することなく使用することを特徴とする。

【0017】また、本発明のファイルアクセス制御方法は、一つのプロセスがCPUへの割り付け単位となるスレッドを複数もつことのできるマルチスレッド環境を提供する計算機システムであって、これらのスレッドそれぞれがファイルそれぞれに固有に割り当てられたファイル記述子を用いてファイルへのアクセスを行なう計算機システムのファイルアクセス制御方法において、前記プロセスがファイルのオープンを要求したときに、そのオープンの要求されたファイルにファイル記述子を割り当てて、そのファイル記述子を前記要求元のプロセスに返送するステップと、前記プロセスのいずれかのスレッドが前記割り当てられたファイル記述子を用いてそのファイルへのアクセスを初めて要求したときに、アクセスポイントを含むそのファイルに関するアクセス制御情報を保持するアクセス制御ブロックをそのスレッドに対応させて前記メモリに確保するステップとを具備し、前記ファイル記述子をプロセス単位の資源として扱うとともに、前記スレッドそれぞれが個別にアクセス制御ブロックをもつことによって、同一プロセス内のスレッドすべてが前記ファイル記述子を排他することなく使用することを特徴とする。

【0018】本発明によれば、プロセスがファイルのオープンを要求してきたときに、システムは、そのファイルにファイル記述子を割り当てるとともに、そのファイル記述子を要求元のプロセスに返送する。したがって、この時点では、アクセス制御ブロックの確保は行なわれない。

10

20

30

40

50

【0019】次に、プロセス内のスレッドがそのファイル記述子を用いて初めてアクセスを行なった際、システムは、このスレッドに対応させて、そのファイルについてのアクセス制御ブロックを確保する。すなわち、各スレッドは、それぞれ固有のアクセス制御ブロックをもつことになる。

【0020】これにより、ファイル記述子をプロセス単位の資源として取り扱うことができ、かつ各スレッド相互間でこのファイル記述子を排他して使用するということが不要となる。

【0021】

【発明の実施の形態】以下、図面を参照して本発明の一実施形態を説明する。図1は同実施形態におけるファイルアクセスの制御原理について説明するための概念図である。

【0022】同実施形態において、プロセス（AP）9が生成されると、システムは、プロセス制御ブロック（PCB）2およびプロセスアサインテーブル（P-ASGN）3をメモリに確保して、このプロセス制御ブロック2をシステム制御ブロック（SCB）1に登録する。

【0023】このシステム制御ブロック1は、システム上で稼働するプロセス全体を制御するために用いられるものである。また、プロセス制御ブロック2は、プロセス9の状態管理などに用いられるものであり、プロセスアサインテーブル3は、ファイル制御ブロック（FCB）4を参照するときに用いられるものである。なお、ファイル制御ブロックとは、ファイルそれぞれに対応して必要に応じてメモリに確保されるものであり、ファイルの属性などを含む制御情報を保持するものである。

【0024】また、プロセス9によりスレッドが生成されると、システムは、スレッド制御ブロック5a～5cおよびスレッドアサインテーブル6a～6cをメモリに確保して、このスレッド制御ブロック5a～5cをプロセス制御ブロック2に登録する。なお、このスレッド制御ブロック5a～5cおよびスレッドアサインテーブル6a～6cは、それぞれ一対にして確保され、各スレッド制御ブロック5a～5cは、それぞれに対応したスレッドアサインテーブル6a～6cのアドレスを保持している。

【0025】次に、このプロセス9がファイル8のオープンを要求すると、システムは、そのファイル8に対応したファイル制御ブロック4のアドレスをプロセスアサインテーブル3のいずれかのエントリに設定し、この設定したエントリをファイル記述子としてプロセス9に返送する。

【0026】そして、プロセス9のスレッドが、このファイル記述子を用いてファイル8へのアクセスを要求すると、システムは、このファイル記述子が、そのスレッドに対応して設けられたスレッドアサインテーブル（こ

こでは6aとする）内で有効であるかどうかを判定する。

【0027】このスレッドアサインテーブル6aの検索は、以下の手順で行なわれる。すなわち、スレッドからのアサイン要求が発生すると、システムは、プロセスIDとスレッドIDとを取得して、まずプロセスIDからプロセス制御ブロック2を知得する。次に、このプロセス制御ブロック2とスレッドIDとからスレッド制御ブロック5aを知得する。そして、このスレッド制御ブロック5aの保持するスレッドアサインテーブル6aのアドレスによりスレッドアサインテーブル6aを検索する。

【0028】そして、システムは、この検索したスレッドアサインテーブル6aにアクセスの要求されたファイル8に対応したアクセス制御ブロック7aが存在するかどうかを検査することにより、そのファイル記述子が、そのスレッドアサインテーブル6a内で有効であるかどうかを判定する。

【0029】なお、このアクセス制御ブロックとは、有編成ファイルにおけるアクセスポインタやレコードロック情報、無編成ファイルにおけるカレントアクセス位置などといったアクセス制御情報を保持するものであり、各ファイルに対応して確保されるものである。また、アクセス制御ブロックは、ファイル制御ブロックを参照するためのアドレスなども保持している。

【0030】ここで、このファイル記述子が有効であった場合には、そのスレッドアサインテーブル6aに設定されたアクセス制御ブロック7aに基づいて、アクセス処理を実施する。一方、ファイル記述子が有効でなかった場合には、さらにそのファイル記述子がプロセスアサインテーブル3内で有効であるかどうかを判定する。

【0031】ここで、このファイル記述子がプロセスアサインテーブル3内で有効でない場合には、ファイルのオープンが行なわれていないこととなるため、エラーとしてプロセス9にその旨を返答する。

【0032】一方、このファイル記述子がプロセスアサインテーブル3内で有効であった場合には、アクセス制御ブロック7aをメモリに確保してスレッドアサインテーブル6aに設定するとともに、そのアクセス制御ブロック7aを用いてアクセス処理を実施する。

【0033】すなわち、各スレッドがアクセス制御ブロックをそれぞれにもつために、ファイル記述子をプロセス単位の資源として扱うことができ、かつファイル記述子の排他をまったく意識する必要がない。

【0034】次に図2乃至図5を参照して同実施形態の動作手順を説明する。図2は同実施形態のプロセスが生成されたときの動作を説明するフローチャートである。

【0035】プロセス9が生成されると、システムは、プロセス制御ブロック2をメモリに確保して（図2のステップA1）、ついでプロセスアサインテーブル3をメ

メモリに確保する(図2のステップA2)。

【0036】図3は同実施形態のプロセスによりスレッドが生成されたときの動作を説明するフローチャートである。プロセス9によりスレッドが生成されると、システムは、スレッド制御ブロック5a~5cをメモリに確保して(図3のステップB1)、ついでこれらスレッド制御ブロック5a~5cそれぞれと一対にしてスレッドアサインテーブル6a~6cをメモリに確保する(図3のステップB2)。

【0037】図4は同実施形態のプロセスがファイルのオープンを要求しときの動作を説明するフローチャートである。プロセス9が、ファイル8のオープンを要求すると、システムは、プロセスIDからプロセス制御ブロック2を知得し、さらにこのプロセス制御ブロック2からプロセスアサインテーブル3を知得する(図4のステップC1)。そして、システムは、このオープン要求のあったファイル8に対応するファイル制御ブロック4がメモリに確保されているかどうかを判定し(図4のステップC2)、確保されていないときに、ファイル制御ブロック4をメモリ上に確保する(図4のステップC3)。

【0038】次に、システムは、プロセスアサインテーブル3のいずれかのエントリを確保して(図4のステップC4)、ファイル制御ブロック4をこのエントリに設定するとともに、プロセス9により指定されたオープンモードを格納する(図4のステップC5)。そして、この確保したエントリのエントリ番号をファイル記述子として、プロセス9に返送する(図4のステップC6)。

【0039】図5は同実施形態のスレッドがファイル記述子を用いてファイルのアクセスを要求したときの動作を説明するフローチャートである。プロセス9のスレッドが、ファイル記述子を用いてファイル8へのアクセスを要求すると、システムは、まずプロセスIDからプロセス制御ブロック2を知得し、さらにこのプロセス制御ブロック2からプロセスアサインテーブル3を知得する(図5のステップD1)。次にシステムは、スレッドIDからスレッド制御ブロック5aを知得し、さらにこのスレッド制御ブロック5aからスレッドアサインテーブル6aを知得する(図5のステップD2)。

【0040】ここでシステムは、ファイル記述子が、そのスレッドアサインテーブル6a内で有効であるかどうかを判定し(図5のステップD3)、有効でない場合には、さらにそのファイル記述子がプロセスアサインテ

\*ブル3内で有効であるかどうかを判定する(図5のステップD4)。

【0041】ここで、このファイル記述子がプロセスアサインテーブル3内で有効でない場合には、ファイルのオープンが行なわれていないこととなるため、エラーとしてプロセス9にその旨を返答する。

【0042】一方、このファイル記述子がプロセスアサインテーブル3内で有効であった場合には、アクセス制御ブロック7aをメモリに確保してスレッドアサインテーブル6aに設定するとともに(図5のステップD5~ステップD6)、そのアクセス制御ブロック7aを用いてアクセス処理を実施する(図5のステップD7)。これにより、ファイル記述子をプロセス単位の資源として扱うことができ、かつファイル記述子の排他をまったく意識する必要がない。

【0043】

【発明の効果】以上詳述したように、本発明によれば、ファイル記述子をプロセス単位の資源として扱えるとともに、スレッドそれぞれが個別にアクセス制御ブロックをもつことによって、同一プロセス内のスレッドすべてがファイル記述子を排他することなく使用することができることとなる。

【図面の簡単な説明】

【図1】本発明の実施形態におけるファイルアクセスの制御原理について説明するための概念図。

【図2】同実施形態のプロセスが生成されたときの動作を説明するフローチャート。

【図3】同実施形態のプロセスによりスレッドが生成されたときの動作を説明するフローチャート。

【図4】同実施形態のプロセスがファイルのオープンを要求しときの動作を説明するフローチャート。

【図5】同実施形態のスレッドがファイル記述子を用いてファイルのアクセスを要求しときの動作を説明するフローチャート。

【図6】従来のマルチスレッド環境下でのファイルアクセスの制御原理を説明するための概念図。

【符号の説明】

1…システム制御ブロック、2…プロセス制御ブロック、3…プロセスアサインテーブル、4…ファイル制御ブロック、5a、5b、5c…スレッド制御ブロック、6a、6b、6c…スレッドアサインテーブル、7a、7b、7c…アクセス制御ブロック、8…ファイル、9…プロセス。



【図6】

